Llombart *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2021) 2021:1
https://doi.org/10.1186/s13636-020-00191-3

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Progressive loss functions for speech enhancement with deep neural networks

Jorge Llombart*, Dayana Ribas, Antonio Miguel, Luis Vicente, Alfonso Ortega and Eduardo Lleida

Abstract

The progressive paradigm is a promising strategy to optimize network performance for speech enhancement purposes. Recent works have shown different strategies to improve the accuracy of speech enhancement solutions based on this mechanism. This paper studies the progressive speech enhancement using convolutional and residual neural network architectures and explores two criteria for loss function optimization: weighted and uniform progressive. This work carries out the evaluation on simulated and real speech samples with reverberation and added noise using REVERB and VoiceHome datasets. Experimental results show a variety of achievements among the loss function optimization criteria and the network architectures. Results show that the progressive design strengthens the model and increases the robustness to distortions due to reverberation and noise.

Keywords: Progressive loss function, Speech enhancement, ResNet, CNN

1 Introduction

Most deep neural network speech enhancement (DNN-SE) methods act like a monolithic block, where the noisy signal is the input to the architecture and the enhanced signal is the output, while intermediate signals are not easily interpretable. However, SE can also be performed as a gradual improvement process, with a step-by-step speech denoising. In this paradigm, the signal is enhanced progressively at different system stages, by incrementally improving the speech quality at each stage in terms of noise reduction, speech distortion, etc.

The incremental SE paradigm has been recently approached through the so-called progressive speech enhancement (PSE) [1–3]. In this mechanism, the network learning process is decomposed in multiple stages, such that the target is progressively optimized. This way, the subproblem solved at each stage can boost the subsequent learning in the next stages. Previous works following this strategy have shown improved results for the progressive architectures compared to usual DNN-SE methods.

*Correspondence: jllombg@unizar.es
ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, C/ Pedro Cerbuna 12, 50009 Zaragoza, Spain

Previous progressive proposals have focused on the incremental signal-to-noise ratio (SNR) reconstruction at different degrees. In [2], a feedforward deep neural network implemented a regression scheme, where the network target was learning an ideal binary mask responsible for improving the SNR three times in 10 dB. The same example was used with different SNR to achieve the progressive enhancement. In [3], the authors extended this work by testing more advanced architectures. Initially, a reproduction of the procedure in [2] using a long short-term memory cell (LSTM) showed a degradation of the SE performance with the number of target layers. Then, at each cleaning step, they used additional knowledge from the previous steps, finally achieving an improvement in performance.

More recently and motivated by the interpretability of the enhancement process, we have presented a progressive architecture based on wide residual networks [1]. Our main goal was to understand the enhancement process, step by step, by using a visualization probe at each network block. Insights provided by the interpretation of the enhancement process led to the modification of the network architecture, which provided improved results for the SE process. In the proposed architecture, the mean

square error (MSE) of the log-spectral amplitude (LSA) between the enhanced signal and the reference is computed at every network stage and refreshes the backpropagation gradients. Furthermore, the reconstruction error of each block contributes to the optimization loss function with a weighted progressive mechanism.

Our preliminary approach to this problem had the intention of just presenting a progressive approach for DNN speech enhancement [1]. Now, this work deeply studies the progressive strategy for DNN-SE. This paper explores the generalization of the training method on two consolidated DNN architectures used for SE tasks: a convolutional neural network (CNN) and a residual neural network (ResNet). This study analyzes two different criteria to implement the progressive paradigm: the weighted progressive (WP) criterion in [1] and a newly proposed uniform progressive criterion (UP). The UP criterion implements the final optimization of the loss function, considering that the reconstruction errors from all blocks contribute in the same way. Moreover, in this work, we consider not only the dereverberation problem but the whole enhancement problem. Also, a wider experimental setup is implemented, including simulated and real datasets.

More recent DNN architectures used for SE such as generative adversarial networks (GAN) [4], U-Net [5], or residual hourglass recurrent neural networks (RHR-Net) [6] have demonstrated their capabilities and currently they offer the best results. Despite these architectures could also benefit from the use of the proposed method, in this work, we concentrate on the performance on a selected set of very well-known, simple, and established architectures to show the benefits in terms of performance without negligible increase in computational complexity (very reduced at training time and no computational increase at inference time) of the progressive approach disregarding the specific method or network architecture.

The contributions of this work are:

- Study of the PSE on two consolidated deep neural network (DNN) architectures: CNN and ResNet.
- Assessment of two criteria for progressive loss function optimization: weighted and uniform.
- Exploring the space of input features.
- Analysis of the progressive mechanism effect on gradients and speech quality measures.

The rest of the paper is organized as follows. Section 2 summarizes the antecedents of this work. Section 3 goes deeper into the application of the progressive paradigm to the loss function. Section 4 describes the experimental conditions. Section 5 presents some preliminary results on the vanishing gradient problem, and Section 6 analyzes the behavior of the CNN/ResNet architectures when they are using the progressive paradigm by presenting obtained results. Finally, Section 7 concludes the paper.

2 Antecedents

The architectures considered in this work are CNN and ResNet. In order to adapt these architectures to the progressive paradigm, it is necessary to add additional restrictions and modify the loss function. In the following subsections, we provide an overview of the architecture design and the loss function that will be the base of this work.

2.1 Architecture

Architectures based on CNN are capable of exploiting local patterns in the spectrum from both frequency and temporal domains [7, 8]. The effect of noise and reverberation appears as a perturbation of the signal spectral shape extended through a specific time-frequency area. The natural structure of the speech signal or the distortion patterns can show correlation in consecutive time-frequency bins in a context. CNN-based architectures effectively deal with this characteristic of the speech signal structure, what makes them appropriate for speech enhancement purposes. CNN has also appeared combined with recurrent blocks to further model the dynamic correlations among consecutive frames [9]. In Fig. 1, we show a typical structure of a CNN where each architecture block could have different configurations in terms of convolutional layers, batch normalization, or non-linearities.

The incorporation of residual connections brought a regularization potential to the CNN approach [10]. ResNet architecture makes use of shortcut connections between neural network layers, allowing systems to handle more depth, with faster convergence and a smaller gradient vanishing effect. Since they can manage deeper networks, they can be more expressive, provide more detailed representations of the underlying structure of the corrupted signal and manage longer contexts. All of this results in more accurately enhanced speech. We show this modification in Fig. 2, where we describe the connection between convolutional blocks in a residual approach.

In [1], we added to the ResNet an additional constraint: the architecture kept a constant number of channels along all the blocks of the DNN. The constant number of channels allowed the output reconstruction and a visualization probe at any internal block. The mandatory progressive signal reconstruction forced an incremental process of the SE that tended to improve the robustness of the model. Besides, this architecture uses a weighted composition of reconstruction errors by block to perform the loss function optimization. This way, each block makes partial reconstruction, and the next block has as input a previously enhanced representation of the signal.

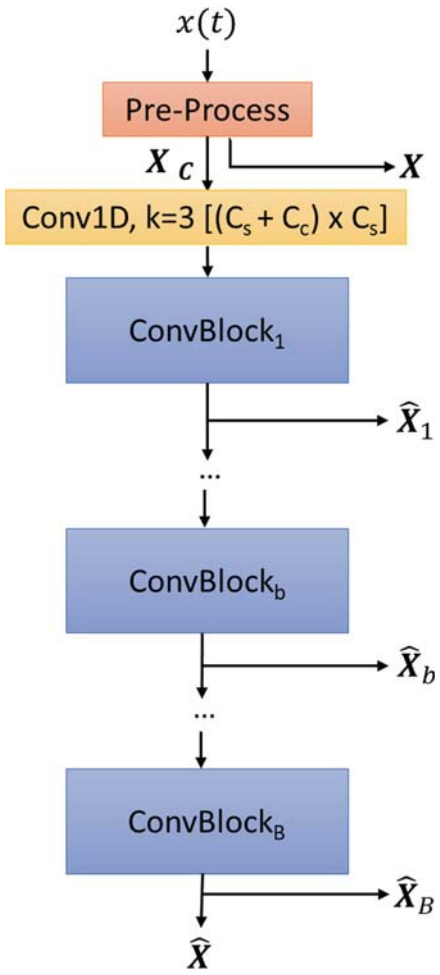

Fig. 1 Convolutional neural network (CNN). The convolutional block can have any distribution of convolutional layers and auxiliary layers such as the batch normalization, or non-linearities

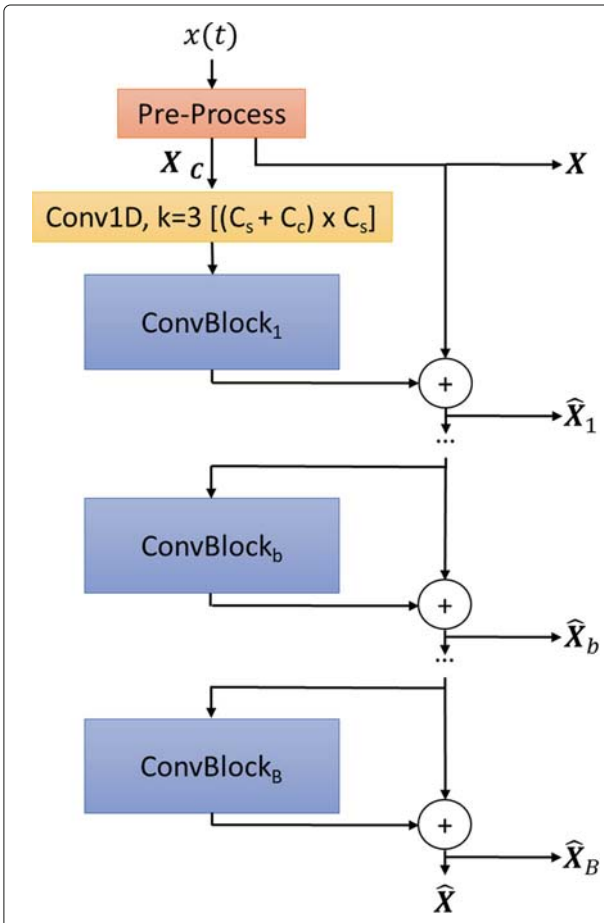

Fig. 2 Residual neural network (ResNet). The convolutional block can have any distribution of convolutional layers and auxiliary layers such as the batch normalization, or non-linearities. The main difference between CNN and ResNet is the residual path in ResNet

2.2 Loss function

In [1], we proposed an SE system based on the reconstruction of the LSA of a noisy signal: the audio signal was reconstructed, by means of the overlap-add mechanism, using the enhanced logarithmic output spectrum with the phase of the original noisy speech. The loss function was the classical MSE between the LSA of the reference and the LSA of the enhanced signal,

$$MSE\left(\boldsymbol{y}_{n,\tau},\hat{\boldsymbol{x}}_{n,\tau}\right)=\frac{1}{D}\sum_{d=0}^{D-1}\left(y_{d,n,\tau}-\hat{x}_{d,n,\tau}\right)^2 \tag{1}$$

where D is the signal input dimension, $\boldsymbol{y}_{d,n,\tau},\hat{\boldsymbol{x}}_{d,n,\tau}$ are the frequency bins of the logarithmic spectrum at the training example n and frame τ. $\boldsymbol{y}_{n,\tau}$ is the target vector of the clean LSA reference, and $\hat{\boldsymbol{x}}_{n,\tau}$ is the reconstructed vector of the enhanced signal.

From our previous experience [1, 11, 12], instead of using a frame-by-frame loss function, this loss uses the whole input as a sequence. Namely, the base loss function is the MSE of the LSA over all the examples and sequence length of an update step,

$$J\left(\boldsymbol{Y},\hat{\boldsymbol{X}}\right)=\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{T}\sum_{\tau=0}^{T-1}MSE\left(\boldsymbol{y}_{n,\tau},\hat{\boldsymbol{x}}_{n,\tau}\right) \tag{2}$$

where $\boldsymbol{Y}$ and $\hat{\boldsymbol{X}}$ are the LSA representation of the training update.

Each example is a sequence of all the frames of the input signal, where N is the number of examples in the training procedure step, and T is the number of frames of the example. In order to simplify the training procedure, all the training examples have the same number of frames. Therefore, the training keeps fixing the same segment size, which is obtained by randomly cropping the input signals. This way, any example selected for a training update is an arbitrary segment of the input example.

Finally, [1] implements the progressive paradigm modifying the objective loss function composing the MSE

between noisy input LSA and the enhanced LSA at different network levels or blocks. This progressive loss function is a particular case of this paper proposal, and it will be studied in detail in the following section.

3 Speech enhancement

This paper aims to study the underlying potential of the PSE paradigm. Previous works have pointed out the performance improvement of the SE task in progressive architecture designs. Beyond these results, this paper brings the hypothesis that the progressive paradigm obtains better SE performance because these mechanisms also refresh gradients during the neural network training. In the following, we will describe the PSE architecture proposed in this paper, which is based on our previous work [1], but additionally includes a set of novelties/contributions designed explicitly for this study.

3.1 Architecture

This paper study will be based on two DNN architectures: progressive convolutional neural network (P-CNN) and progressive residual neural network (P-ResNet). Beyond our previous proposal in [1] using the ResNet topology, this paper includes the CNN topology with comparative purposes and to extend the study to generalize the progressive paradigm to different architectures.

Figure 3 represents the front-end of both architectures. The input signal, $x(t)$, is first windowed, and then, we obtain the logarithm of the absolute value of its short-term Fourier transform (STFT), yielding the LSA $\boldsymbol{X}$. We also obtain the Mel-scaled filter bank (FB), and Mel-frequency cepstral coefficients (MFCC) with different windowing processes to provide additional information to the network, $\boldsymbol{X}_C$.

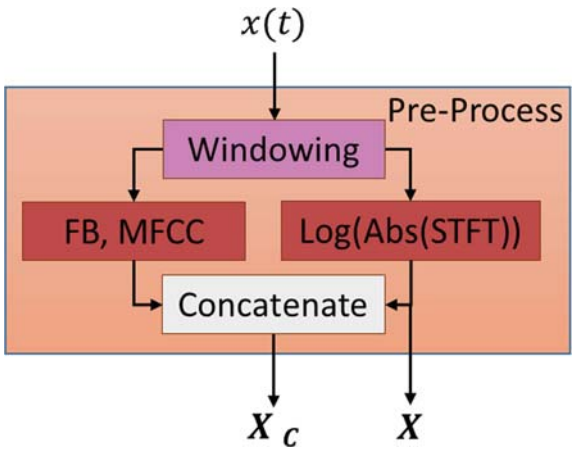

Fig. 3 Front-end. Preprocessing block at beginning of P-CNN and P-ResNet architectures

Both architectures keep the same number of channels along all their convolutional blocks. Also, they use the same basic convolutional block (Fig. 4) to remain as comparable as possible. This convolutional block is composed of two successive identical structures. This structure starts with batch normalization, followed by a parametric rectified linear unit (PReLU), and a 1D-Convolutional layer with the same number of channels at the input and the output. In Fig. 4, C_s is the number of channels. The dimension of the kernel (k) is 3 in all convolutions of the architecture. The output of this structure has the same dimensions as the enhanced output. Thus, we can obtain a partially enhanced signal at each block output of P-CNN and P-ResNet.

For this work, we used 1D-convolutional layers. Unlike 2D-convolutional layers that combine temporal and frequency dimensions locally, 1D-convolutional layers perform a global combination over all the frequency dimensions in a short-term temporal context. Recent works suggest that when convolutional architectures are employed,

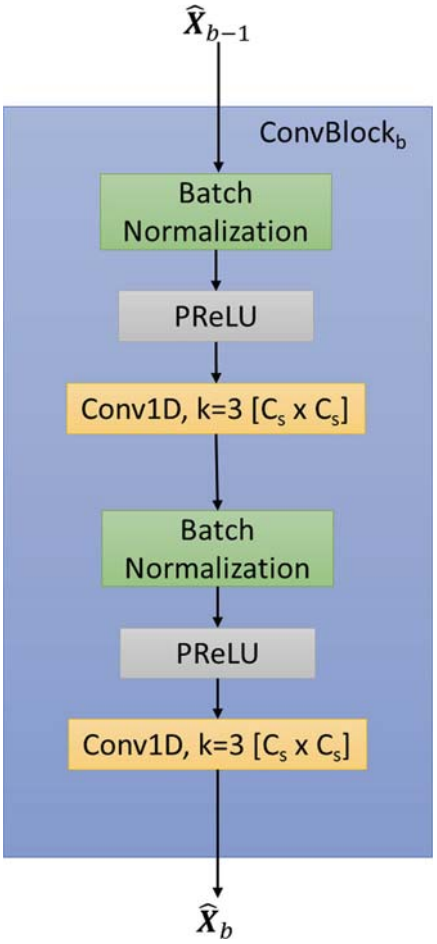

Fig. 4 Convolutional block. Basic block for composing P-CNN and P-ResNet architectures

the use of convolutional layers computed through the single temporal dimension are more appropriate for the speech enhancement processing [13, 14].

The multiresolution windowing processing of the signal contributes to the dereverberation task, especially when the impulse response is longer than the window length used in the LSA analysis [15]. $\boldsymbol{X_c}$ output is only used as input to the first convolutional block as shown in Figs. 1 and 2. The following blocks have the same input and output dimensions to allow the use of the progressive loss function. By providing the MFCC, the network has the possibility of calculating average cepstral representations to help with the channel identification and improve the dereverberation. The filter bank can also play a role in the identification of useful speech structures in a perceptual scale. As we can see in the experiments, their combined use makes a significant improvement.

3.2 Loss function

In [1], we designed a neural network to have the same number of channels as the input signal at certain probe points. To induce the desired behavior, we forced the desired enhanced signal to be obtained at these points by adding their reconstruction errors to the training loss, which provided a progressive reduction of the difference between the reference signal and the reconstruction after each block. Unlike the classical layer-wise training, where a stacking technique is used, we train the whole network against the final objective in the proposed method but with the additional constraint that a full reconstruction after each architecture block must be carried out. Our previous work demonstrated that if we do not force the reconstruction after each block, intermediate block outputs are entirely different from our objective and not interpretable. The inclusion of the reconstruction constraint through our loss function allows the visualization of the enhancement procedure. We can choose an intermediate result to reduce the evaluation computational cost depending on the application and help the training procedure to obtain better results.

With the proposed loss function, we add the full reconstruction constraint after each convolutional block minimizing the MSE between the clean reference $\boldsymbol{Y}$ and the block output $\hat{\boldsymbol{X}}_b$ (Fig. 5). Equation 3 shows a general definition of the progressive loss function as a weighted sum over the reconstruction loss of each convolutional block

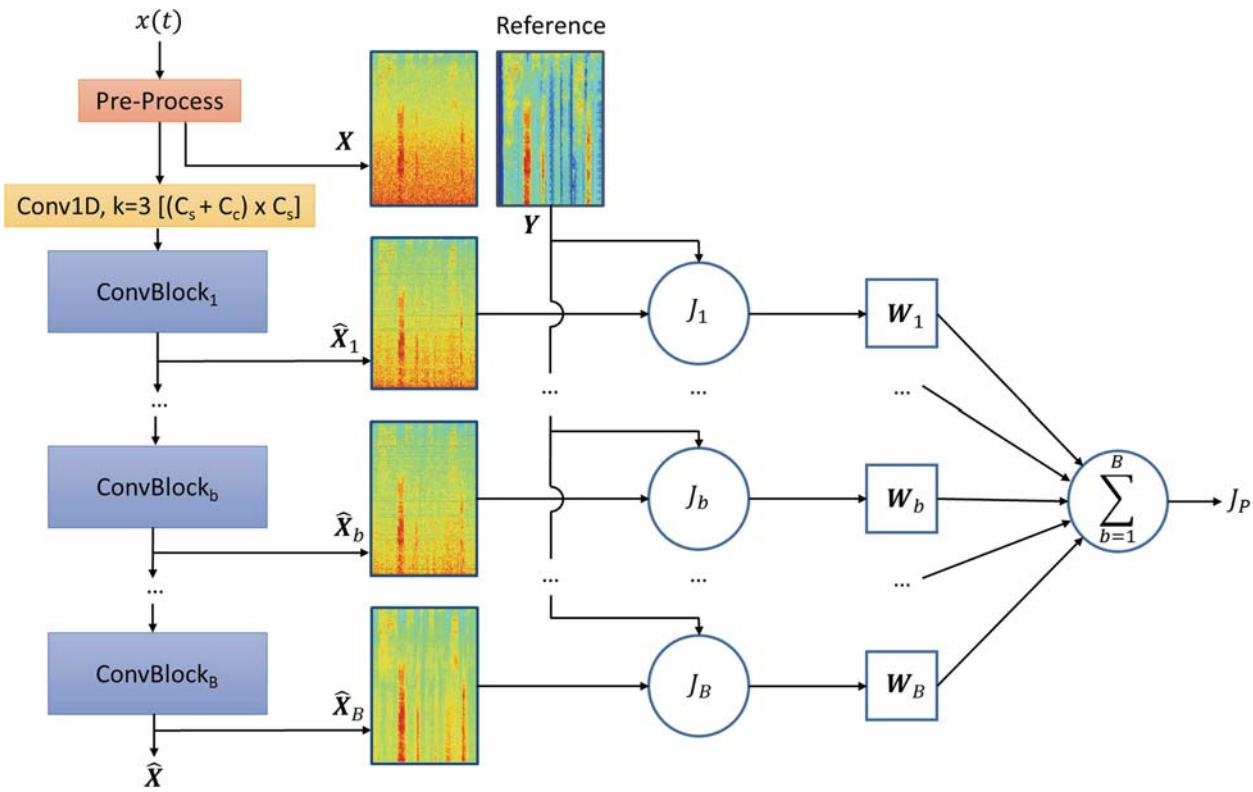

Fig. 5 PSE general architecture for P-CNN and P-ResNet. This figure illustrates the application of the progressive loss that allows to directly represent the output after each block

Table 1 Training datasets description

Dataset	Timit	Librispeech	TedLium
Files	6299	292329	56704
Speakers	630	2484	698
Speech type	Read speech		Conference
Interface	Close microphone		Auditorium microphone

$$J_P\left(\boldsymbol{Y},\hat{\boldsymbol{X}}\right)=\sum_{b=1}^{B}W_b\cdot J\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_b\right). \tag{3}$$

Depending on the weights in Equation 3, it is possible to define different progressive loss function criteria. In [1], we proposed the WP loss function and here we also propose the UP criterion. In the next sections, both criteria are experimentally evaluated in combination with P-CNN and P-ResNet.

- Weighted progressive (WP): The main weight of the loss function is the final cost, as usual in approximation tasks. Then, the cost of all the architecture blocks is uniformly distributed and added in a weighted sum,

$$J_{WP}\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_B\right)=J\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_B\right)+\alpha\frac{1}{B}\sum_{b=1}^{B}J\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_b\right) \tag{4}$$

  where B is the number of blocks of the architecture. Note that Equation 4 is a particular case of the general progressive loss function in Equation 3, where $W_b=\alpha/B$ for $b=1,\ldots,B-1$ and $W_B=1+\alpha/B$. This loss function implements progressive processing along blocks, i.e., every intermediate block reconstructs the enhanced signal. This design forces the enhancement process to be incremental, from slightly to detailed cleaning. In the end, this processing complements the traditional process to obtain the final system output, namely the standard back-propagation of gradients throughout the full architecture (output-input).
- Uniform progressive (UP): This loss function proposes a uniform distribution of the block losses along the architecture,

$$J_{UP}\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_B\right)=\frac{1}{B}\sum_{b=1}^{B}J\left(\boldsymbol{Y},\hat{\boldsymbol{X}}_b\right), \tag{5}$$

  which is a special case of Equation 3 where $W_b=1/B$ for $b=1,\ldots,B$.
  With this strategy, all the outputs have the same impact in the reconstruction. This way, every block can equally contribute to the final loss, and the full architecture makes the same effort in the signal reconstruction.

4 Experimental setup

4.1 Training data

For DNN training, we have used three different public datasets: Tedlium [16] from Ted talks; Librispeech [17], audio-books; and Timit [18], a phonetically balanced distributed read speech. These datasets are fully employed, without any partition. See Table 1 for the characteristics of the datasets.

4.2 Data augmentation: reverberated and noisy training data

Data augmentation using reverberation and additive noise was performed at the training set. For each random training example, there are three transformations (See Table 2 for further details):

1. Impulse responses: We simulated random rooms and source-receiver distances described through the room impulse responses (RIR) using the python package rir-generator[1] [19]. For the data augmentation loop, there are three different kinds of simulated rooms: small, medium, and large, selected with a probability of 0.5, 0.3, and 0.2.
2. Additive noise: We add some noise, with SNR uniformly sampled between 5 and 25 dB, from the music and noise files in the Musan dataset [20]. Note that among the noise files, there is crowd noise, but there is not any intelligible speech.
3. Time scaling: We randomly select a scale between 0.8 and 1.2. There are signals with no scaling, i.e., the original speed. Some others are slowed down or sped up.

4.3 Evaluation data

For evaluation purposes, we use two databases: (1) REVERB [21] and (2) VoiceHome v0.2 [22] and v1.0 [23]. REVERB is divided in a development set (REVERB-Dev), generally used for evaluating intermediate results during the study, and an evaluation set (REVERB-Eval), for confirming the results and evaluation of the system. VoiceHome evaluates the system in a realistic domestic environment with noise and reverberation. So, with these two databases, we can separate two conditions:

[1] https://github.com/Marvin182/rir-generator

Table 2 RIR and noise for training data augmentation

	Room impulse responses		
	Small	Medium	Large
Probability	0.5	0.3	0.2
Size (x,y,z)[m]	$x \sim U(1,6), y \sim U(1,6), z \sim U(2,3.5)$	$x \sim U(6,10), y \sim U(6,10), z \sim U(3,5)$	$x \sim U(10,20), y \sim U(10,20), z \sim U(4,6)$
RT_{60}[s]	$RT_{60} \sim U(0.1,0.25)$		
Distance[m]	0.5, 1.0, 1.5, 2.0, 2.5		
Microphone type	Bidirectional, hypercardiodid, cardioid, subcardoid, omnidirectional		
	Noise		
Music	659 files		
Noise	929 files		
SNR [dB]	$SNR \sim U(5,25)$		

Simulated data Part of the REVERB dataset corresponds to simulated conditions. They are speech samples from the WSJCAM0 corpus [24] combined with three kinds of RIR: small, medium, and big room ($RT_{60} = 0.25, 0.5, 0.7s$). For each one, there are two source-mic distances: far ($2m$) and near ($0.5m$). Also, a stationary noise was added from the same rooms ($SNR = 20dB$). For this study, we only use the first channel of the eight available. We also add five noises ($SNR = 0, 5, 10, 15, 20$, and $25dB$) to all signals at the simulated condition of REVERB. These are babble noise, cafe environment noise, music, street environment with lot of traffic, and noise captured inside a moving tram.

Real data We used two evaluation sets with real conditions: the real part in REVERB and VoiceHome dataset (v0.2 and v1.0). REVERB was recorded in a meeting room with $RT_{60} = 0.7s$ at two distances: far ($2.5\ m$) and near ($1\ m$), from MC-WSJ-AV [25]. VoiceHome corresponds to a realistic domestic environment with everyday noises like a vacuum cleaner, dish-washing, or sound of TV shows.

4.4 Speech quality measures

To measure the level of denoising and dereverberation achieved by the PSE method, we estimate the segmental SNR [26] and the speech-to-reverberation modulation energy ratio (SRMR) [27, 28]. In these metrics, the higher the values, the better speech quality. However, it is well-known that the SE processing might generate distortion on the output speech. Therefore, for the simulated dataset, we also measure the distortion between the clean reference and enhanced speech using the log-likelihood ratio (LLR) [29]. In this case, lower values mean less distortion, so the better quality of the speech. The combination of both speech quality viewpoints, i.e., the trade-off between noise/reverberation reduction and distortion, provides a general assessment of the SE method performance. This way, the best enhancement system is the one which improves SNR or SRMR, but retains the distortion, in this case, measured with LLR, as low as possible. Additionally we use the well-known PESQ measure [30] for simulated data. PESQ measure is in range 0-5 where the higher the better performance.

4.5 Neural network configuration

The input provided to the CNN, ResNet, P-CNN, and P-ResNet architectures consists of the logarithm of the magnitude of the 512-STFT of the corrupted signal, sampled at 16 kHz. The STFT is computed every 10 ms for a 25 ms sliding Hamming window. We also concatenate the Mel-Scaled Filter-bank and the MFCC as auxiliary inputs, with filter bank sizes 32, 50, and 100, every 10 ms. MFCC are computed using the discrete cosine transform (DCT) without truncation. However, each frequency resolution has a different sliding Hamming window of 25 ms, 50 ms, and 75 ms respectively. These auxiliary features provide different frequency and temporal resolutions, which can benefit the speech enhancement process [15]. Taking into account that the LSA dimension is 512, the overall input size is 876.

For all the experiments, we use adaptive moment estimator (Adam) as the update function. Each layer has 512 neurons to follow the philosophy of maintaining unaltered the number of channels along the architecture. The training consists of 900 epochs. For each epoch, 10,000 input files are randomly selected from the training set. As long as there are unused training examples, no file can be selected more than once. Batch normalization moving parameters are blocked after epoch 700. For the J_{WP} loss function, we use $\alpha = 0.1$ as in [1], which provided the best SRMR value on REVERB-Dev.

5 Preliminary gradient study

This section presents a preliminary study of the behavior of the gradient to explore how the injection of new fresh gradients at different architecture levels improve

the training procedure. When gradients back-propagate through a large number of layers, they tend to lose energy. Thus, their ability to move weights of the layers near to the input is reduced. The proposed PSE method feeds a fresh and stronger gradient after each block to move the weights of each layer. In order to check this, we design an experiment to observe the energy of the gradients that modify the weights of the first convolutional block during the 100 first optimization updates. This procedure is repeated 100 times with different weight initializations to observe the variance among different starts and the variation of this gradient energy during optimization.

Figure 6 presents the results obtained for P-CNN and P-ResNet architectures, for non-progressive baselines, and for each proposed progressive criteria. There is a noticeable difference in the behavior of the two structures. In P-CNN, there is a significant difference among the gradient energy of each compared system. The lower energy corresponds to the baseline architecture, the one without any progressive assumption. On the other hand, the progressive mechanisms show a significant lifting of the gradient energy. These boosted gradients have more strength to move the weights allowing a better learning at inner layers of the whole architecture.

In contrast, in P-ResNet, there is no relevant difference between the gradient energy of the progressive techniques and that of the no progressive baseline at the first convolutional block. Consider that P-ResNet is an architecture designed to deal with the vanishing problem, and thanks to residual connections, the gradients have a shortcut to propagate up to the first layers without vanishing. In this case, injecting new gradients does not push much more the previous gradients. However, the new gradients are more accurate because they directly come from the target evaluation at the output of each architecture block.

6 Results and discussion

6.1 Analysis of alternatives for the DNN input

In this section, we present a study to asses that the combined use of complementary inputs to the corrupted LSA may improve the performance of the system. We use multiresolution in the MFCC and FB inputs as described in Section 4.5, but we perform an ablation study about the use of each feature type. For this study, we focus on the dereverberation performance of the P-ResNet with

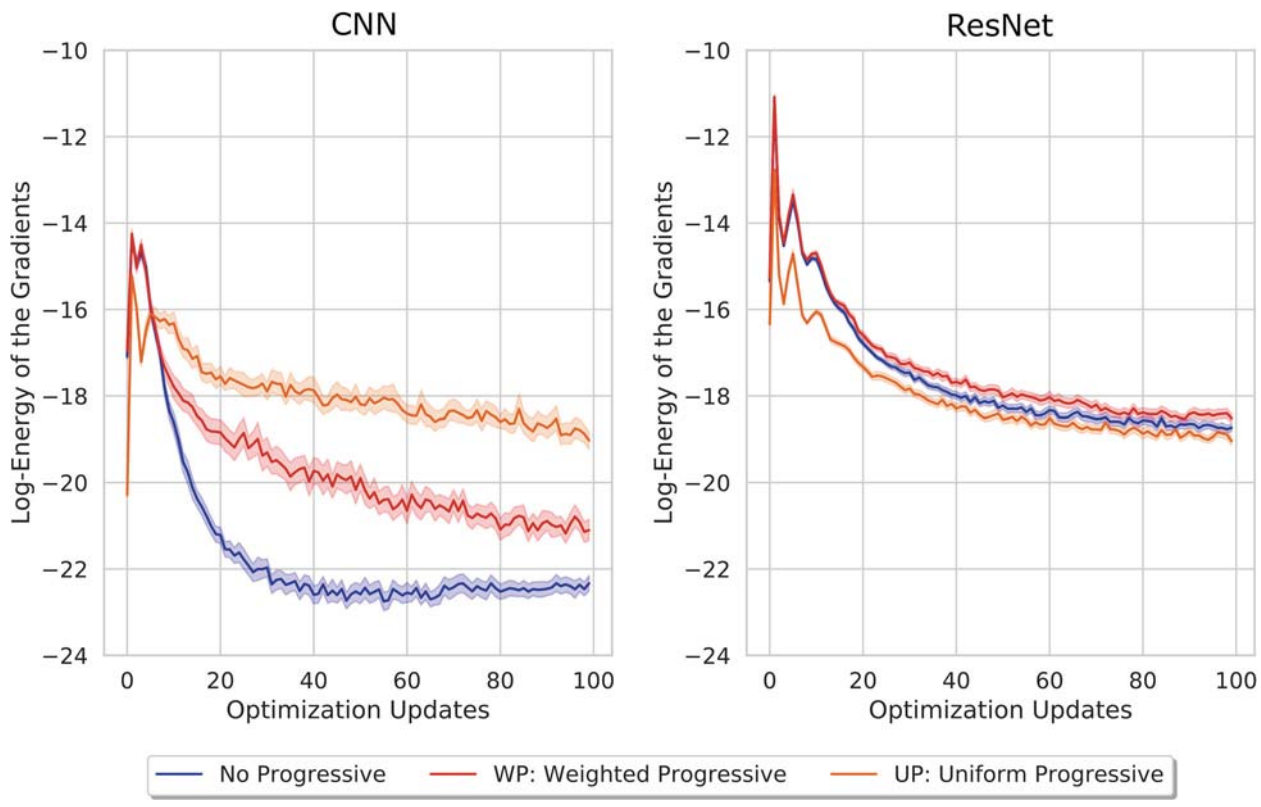

Fig. 6 Mean and standard deviation (shaded area) of the log-energy of the gradients of 100 random network initialization during the 100 first mini-batches in training. The log-energy is measured on the nearest block to the input

Table 3 Evaluation of the use of complementary information at the input of the P-ResNet with WP architecture over the REVERB-Dev dataset in terms of reverberation measured with SRMR

Complementary info.	Real condition	Simulated condition
Without	6.55	7.99
FB	**7.25**	8.31
MFCC	7.07	**8.44**
FB + MFCC	*7.14*	*8.41*

Bold text remarks on the best result per condition and italic text the second best

WP over the REVERB-Dev dataset in real and simulated conditions.

Table 3 shows that the best results in simulated conditions are attained using only MFCC, but for real conditions they are obtained with FB features. On average, the combined use of both features, FB and MFCC, provides the best performance, especially compared to the use of LSA without any auxiliary inputs.

6.2 Architecture depth analysis

SE progressive methods use a sequence of steps to perform the enhancement. We have to determine the number of steps or the number of blocks that composes the architecture. Table 4 shows the architecture depth study in terms of SRMR over the REVERB-Dev dataset. This study shows the results for simulated and real conditions and the average of both.

Result indicate that the configuration with 16 blocks achieves the best performance for all the evaluated conditions. Note how progressive systems can achieve high SRMR, both for simulated and real conditions. This consistency among different conditions demonstrates how the progressive strategy can provide a better generalization to the DNN training.

For CNN topology, the reference system in real conditions quickly degrades the performance with the depth of the architecture. Besides, results for P-CNN with UP are better than the CNN reference system, i.e., P-CNN with UP does not degrade as fast as CNN reference system as depth increases.

For ResNet topology, the availability of residual connections works well with a high number of blocks. For instance, the results of the ResNet reference system achieve the best performance on simulated conditions with the deeper architecture (32 blocks). However, note that in real conditions, the ResNet reference system achieves the best result with 8 blocks versus the 32 blocks for simulated conditions. Nevertheless, P-ResNet

Table 4 Speech quality in terms of SRMR for simulated and real reverberated speech samples through architecture depth for REVERB-Dev dataset. The last rows represents the mean and standard deviation along the experiments presented for each column

		Reference systems		Progressive systems			
Condition	Blocks depth	CNN	ResNet	P-CNN with WP	P-CNN with UP	P-ResNet with WP	P-ResNet with UP
Simulated	8	7.33	8.23	6.49	7.53	8.31	7.91
	16	7.60	8.27	**8.96**	7.70	8.41	8.05
	24	8.87	8.14	6.18	8.09	8.03	8.02
	32	7.01	8.56	7.65	7.41	7.98	7.78
Real	8	6.05	6.82	4.90	6.32	7.06	6.91
	16	5.98	5.81	3.74	**7.26**	7.14	6.85
	24	4.76	5.77	2.07	6.90	6.53	6.91
	32	3.35	6.33	2.33	6.34	5.97	6.62
AVG5±STD	8	6.69±0.64	7.52±0.70	5.69±0.79	6.92±0.60	7.68±0.62	7.41±0.50
	16	6.79±0.81	7.04±1.23	6.35±2.61	7.48±0.22	**7.77±0.63**	7.45±0.60
	24	6.81±2.05	6.97±1.16	4.12±2.05	7.49±0.59	7.28±0.75	7.46±0.55
	32	5.18±1.83	7.44±1.11	4.99±2.66	6.87±0.53	6.97±1.00	7.20±0.58

Bold values show the best result for each condition

with WP improves the reference best result in real conditions with 16 blocks, which is also the P-ResNet best configuration in simulated conditions.

6.3 Progressive enhancement along architecture blocks

In this section, we analyze the behavior of the PSE on speech data affected by different reverberation levels. We use signals from large and small rooms from the simulated condition of REVERB-Dev, which provides samples with several room sizes and source-microphone distances.

Figure 7 shows the evolution of the MSE between the clean reference and the reconstruction at each block output for P-CNN and P-ResNet with WP and UP criteria. First, we can observe that the reconstruction error decreases with the distance between source and microphone, i.e., there is less error for samples in the near distance. In near conditions, the source is close to the receiver and the energy of the direct path speech is larger than that of the reverberate path. Therefore, the reverberation effect does not affect considerably to the listener, generating less error at the evaluation.

Concerning the room size, the far distance in a large room achieves the higher errors for all evaluated cases, which is an expected result because this condition presents the highest reverberation level. However, note that for the small room condition, there is not noticeable difference between far and near conditions, since in small rooms the reverberation level is lower.

In relation to the progressive supervision, there is a noticeable drop in the error at the last block in P-CNN with WP. In P-ResNet with WP, there is also some drop in the last block, but the overall enhancement is more distributed among all the blocks. WP is making a great effort in the reconstruction at this last block. Conversely, the reconstruction effort of UP is more gradual and distributed among all the blocks. For P-CNN with UP, the error remains quite stable for all the blocks. In the small room condition, the error increases in the first block until it stabilizes, which could suggest improving the SE performance by reconstructing from the first layer. However, in the big room condition, the error decreases with blocks. In P-ResNet, we can see a constant decrease in error along the blocks as expected.

Results indicate that the use of progressive supervision is favorable to the SE system, even though depending on the architecture, the more suitable progressive strategy can vary. In general, we can conclude that PSE contributes to the neural network results improvement.

Due to the different behavior between real and simulated, we show the average among conditions to see the trend. Once again, we can see that for all progressive systems the best performance is obtained with $B = 16$.

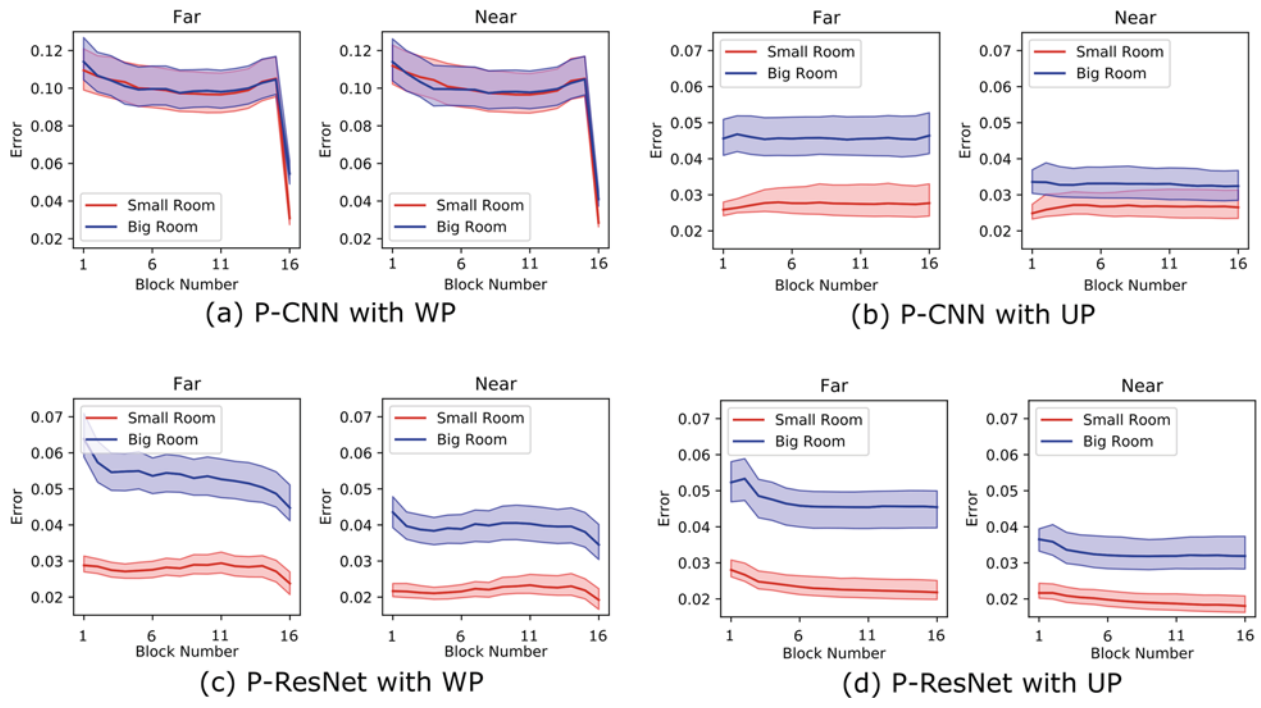

Fig. 7 MSE between clean reference and reconstruction output at each block on the REVERB-Dev set. The dark line shows the mean and the shaded area around shows the standard deviation

Finally, to check the effect of the progressive design directly on the enhancement performance, Fig. 8 shows the evolution of speech quality measures (SRMR, LLR, and PESQ) after each block of the network. Note that when reverberation is highly removed, the distortion can worsen. This indicates that there exists a trade-off between dereverberation and distortion. Finally, PESQ curves show that the overall performance is improved with the block number. The increase of performance is sharp at the last block for WP and smoother for UP.

6.4 Dereverberation

To assess the impact of the PSE proposal in dereverberation tasks, we use SRMR quality measure and LLR for the distortion introduced by the method. This last one only for simulated conditions. Experiments are conducted on REVERB-Eval and VoiceHome v0.2 and v1.0, which also have some noisy conditions. For comparison purposes, we use a DNN variation of the state-of-the-art dereverberation method weighted prediction error (WPE) [31], which uses LSTM [32] .

Table 5 shows the SRMR, LLR, and PESQ results for reference and progressive systems. PSE methods present the best results. In simulated conditions, the best SRMR corresponds to P-CNN with WP, although it also introduces the highest distortion. P-ResNet with WP achieved a bit less SRMR but with less distortion, making it a better speech quality trade-off. We can conclude that the

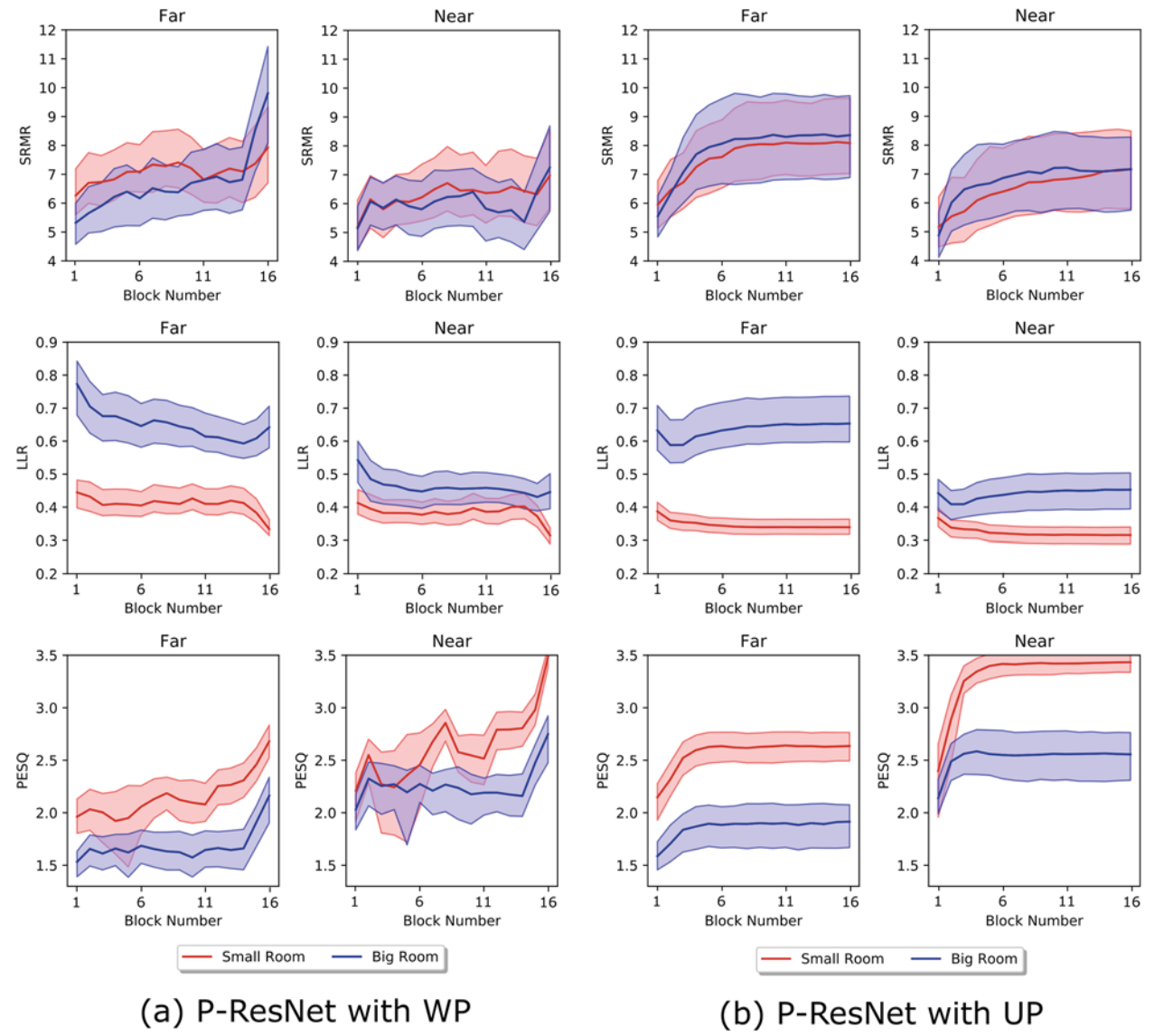

Fig. 8 Evolution of SRMR, LLR, and PESQ for convolutional blocks at P-ResNet with **a** WP and **b** UP on the REVERB-Dev set. The dark line shows the mean and the shaded area around shows the standard deviation

Table 5 Speech quality in terms of SRMR, LLR, and PESQ for simulated and real reverberated speech. The last row represents the mean and standard deviation along the experiments presented in each column

Dataset		Reference systems				Progressive systems			
		Unproc.	WPE [32]	CNN	ResNet	P-CNN with WP	P-CNN with UP	P-ResNet with WP	P-ResNet with UP
Simulated condition									
REVERB	SRMR	6.34	6.64	7.37	7.90	**8.16**	7.46	*8.08*	7.84
Eval	LLR		0.57	0.49	**0.47**	0.79	0.53	*0.48*	0.49
	PESQ		2.27	2.57	*2.68*	2.16	2.43	**2.73**	2.57
Real condition									
REVERB Eval	SRMR	3.44	3.74	5.86	5.79	3.84	**7.23**	*7.00*	6.83
VoiceHome V0.2	SRMR	3.23	3.38	5.80	5.69	2.58	5.49	**7.32**	*5.72*
VoiceHome V1.0	SRMR	4.04	4.47	5.89	6.13	2.78	5.81	**7.31**	*6.27*
Average									
AVG±STD	SRMR	4.26±1.24	4.56±1.26	6.23±0.66	6.38±0.89	4.34±2.26	6.50±0.86	**7.43±0.40**	*6.66±0.78*

Bold results correspond with the best dataset value, and italic results show the second-best value

PSE introduces additional distortion, but it is not significant compared with the performance increase in terms of SRMR. The overall quality represented with the PESQ measure confirms that. Regarding quality and intelligibility measures for simulated conditions, the best results are those of P-ResNet with WP, which obtains the best trade-off between high dereverberation and low distortion.

In real conditions, the best result for the REVERB dataset corresponds to P-CNN with UP, while for the VoiceHome dataset, the best result corresponds to P-ResNet with WP. This last one is the most consistent along the databases because, although for the REVERB dataset was not the best result, P-ResNet with WP is the second-best. The P-CNN with UP has a high discrepancy between simulated and real conditions.

Table 5 also shows the average (AVG) of the evaluated systems for each architecture and its standard deviation (STD). In this case, P-ResNet with WP achieves the best result and with less variability between evaluation datasets. This outcome demonstrates that P-ResNet with WP is the best performing structure. Thus, P-ResNet with WP is the most general-purpose architecture for dereverberation approaches.

6.5 Noise reduction in reverberate environment

This section discusses the performance of the proposed systems on noise reduction using the noisy simulated data on REVERB (see Section 4.3). SNR measures the speech quality performance of SE for denoising level, and LLR, for distortion level. PESQ measures also show the overall quality of speech enhancement.

Figure 9 shows the SNR increase and LLR after speech enhancement (y-axis) versus the initial SNR at the input (x-axis). ΔSNR is the improvement we measure in the estimated output SNR with the Wada method [26] after enhancement, with respect to the input SNR: $\Delta SNR = SNR_{Out} - SNR_{In}$ The results are consistent with previous dereverberation results shown in Section 6.4. For CNN topology, the P-CNN with UP achieves the best outcome, while for ResNet topology, the P-ResNet with WP achieves the best performance.

In evaluation, we used input signals with $SNR = 0$. We did not include this condition in the training procedure, but all the systems obtain an excellent result on enhancement at this point. Moreover, while the input is less challenging, the systems gain in performance until the input is so clean that the systems cannot clean it much more.

In terms of distortion, systems with less LLR are those without progressive supervision, but P-ResNet systems are very close to them. In CNN architecture, UP does not introduce much more distortion than its reference system. Nevertheless, in ResNet architecture, all systems distorted the signal in the same way, although at low input SNR levels, the reference system is the less distorter system.

Table 6 summarizes the results of the noise reduction evaluation, namely the average of ΔSNR, distortion, and PESQ of all noise types and initial SNR for each evaluated system (See the full results in Table 7). The best denoising system is the P-ResNet with WP, followed by P-CNN with UP. These two systems significantly outperform the reference systems of the same architectures, either CNN or ResNet. In the case of P-CNN with WP, there is a huge decrease in performance.

Let us consider now what is the best trade-off in practical terms for SNR-Distortion. The system that introduces

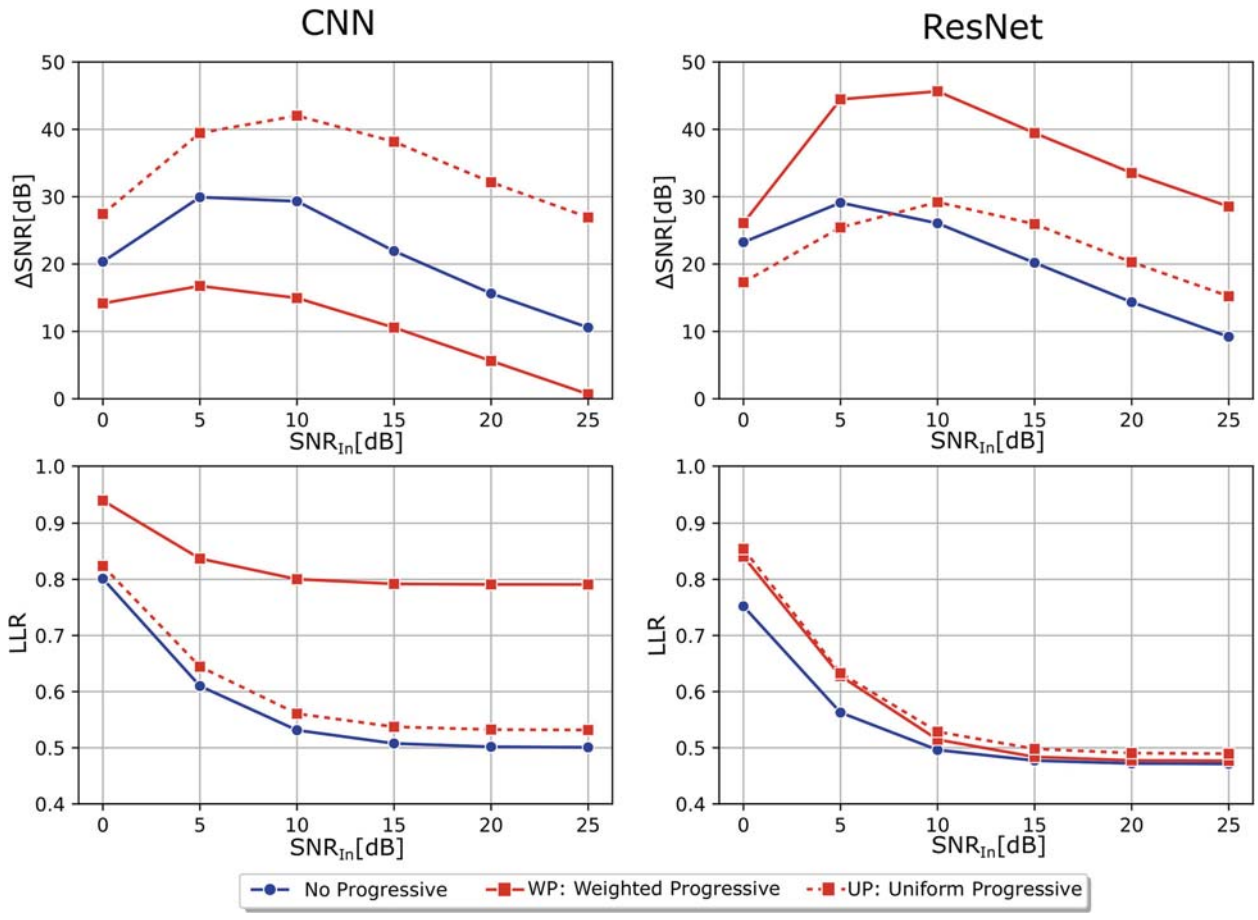

Fig. 9 $\Delta SNR = SNR_{Out} - SNR_{In}$ and LLR after enhancement for both architectures in REVERB-Eval with noise

less distortion is the reference ResNet, but it is also one of the worst at denoising. The second-best system at distortion level is P-ResNet with WP. In addition to that, this is also the best system for denoising tasks. In terms of speech quality, PESQ corroborates that the best system is P-ResNet with WP. Therefore, we can conclude that the progressive strategy also works well for noise reduction, and the system which offers the best trade-off is the P-ResNet with WP.

7 Conclusions

This paper presented a study of PSE, including analysis with CNN and ResNet architectures. Two criteria for progressive loss function optimization have been explored, the weighted and uniform progressive strategies, this last one being a novel proposal. Results have demonstrated that progressive supervision is valuable in both CNN and ResNet architectures. The proposals have achieved an improvement in dereverberation and denoising tasks

Table 6 Summary of speech quality in terms of ΔSNR, LLR, and PESQ for simulated reverberated and noisy speech samples in REVERB-Eval. Mean through all noise types and initial SNR levels conditions evaluated

	Reference Systems		Progressive Systems			
	CNN	**ResNet**	**CNN with WP**	**CNN with UP**	**ResNet with WP**	**ResNet with UP**
ΔSNR [dB]	21.29	20.36	10.46	*34.36*	**36.28**	22.23
LLR	0.58	**0.54**	0.83	0.61	*0.57*	0.58
PESQ	2.24	*2.37*	1.81	2.14	**2.39**	2.21

Bold results correspond with the best dataset value, and italic results show the second-best value

Table 7 Results in simulated REVERB-Eval set for different noises at different initial SNR

SNR_{Input}	Babble	Cafe	Music	Traffic	Tram	Average
CNN (estimated SNR / LLR)						
0	12.74 / 0.84	20.00 / 0.77	13.24 / 0.93	29.68 / 0.71	26.09 / 0.76	20.35 / 0.80
5	29.24 / 0.64	34.23 / 0.60	26.62 / 0.66	42.24 / 0.56	42.28 / 0.59	34.92 / 0.61
10	38.65 / 0.54	40.57 / 0.53	32.45 / 0.54	42.83 / 0.51	42.10 / 0.53	39.32 / 0.53
15	37.14 / 0.51	37.31 / 0.51	33.95 / 0.51	38.47 / 0.50	37.78 / 0.51	36.93 / 0.51
20	35.86 / 0.50	35.76 / 0.50	34.71 / 0.50	35.95 / 0.50	35.88 / 0.50	35.63 / 0.50
25	35.61 / 0.50	35.59 / 0.50	35.37 / 0.50	35.65 / 0.50	35.63 / 0.50	35.57 / 0.50
P-CNN with WP (estimated SNR / LLR)						
0	10.45 / 0.93	15.38 / 0.90	11.11 / 1.06	17.56 / 0.89	16.35 / 0.92	14.17 / 0.94
5	19.18 / 0.83	22.31 / 0.82	18.73 / 0.89	24.51 / 0.81	24.01 / 0.82	21.75 / 0.84
10	24.06 / 0.80	25.32 / 0.80	23.09 / 0.81	26.33 / 0.79	26.00 / 0.80	24.96 / 0.80
15	25.41 / 0.79	25.71 / 0.79	24.77 / 0.79	26.12 / 0.79	25.81 / 0.79	25.57 / 0.79
20	25.66 / 0.79	25.69 / 0.79	25.34 / 0.79	25.74 / 0.79	25.68 / 0.79	25.62 / 0.79
25	25.69 / 0.79	25.70 / 0.79	25.61 / 0.79	25.73 / 0.79	25.70 / 0.79	25.68 / 0.79
P-CNN with UP (estimated SNR / LLR)						
0	14.34 / 0.84	34.01 / 0.78	15.58 / 0.99	31.38 / 0.72	41.89 / 0.78	27.44 / 0.82
5	34.57 / 0.65	51.18 / 0.63	34.37 / 0.71	46.58 / 0.61	55.58 / 0.62	44.46 / 0.64
10	49.16 / 0.57	54.36 / 0.56	47.56 / 0.57	53.47 / 0.55	55.58 / 0.56	52.03 / 0.56
15	52.56 / 0.54	53.52 / 0.54	50.77 / 0.54	54.54 / 0.54	54.40 / 0.54	53.16 / 0.54
20	52.19 / 0.53	52.28 / 0.53	51.36 / 0.53	52.42 / 0.53	52.46 / 0.53	52.14 / 0.53
25	51.99 / 0.53	51.95 / 0.53	51.78 / 0.53	51.98 / 0.53	51.94 / 0.53	51.93 / 0.53
ResNet (estimated SNR / LLR)						
0	12.94 / 0.79	27.71 / 0.73	15.39 / 0.84	27.58 / 0.67	32.64 / 0.73	23.25 / 0.75
5	26.44 / 0.59	38.18 / 0.55	27.15 / 0.59	35.72 / 0.53	43.03 / 0.55	34.11 / 0.56
10	34.51 / 0.51	37.63 / 0.50	31.89 / 0.50	36.97 / 0.49	39.25 / 0.49	36.05 / 0.50
15	34.96 / 0.48	35.46 / 0.48	33.55 / 0.48	36.04 / 0.47	35.94 / 0.48	35.19 / 0.48
20	34.41 / 0.47	34.45 / 0.47	33.86 / 0.47	34.50 / 0.47	34.51 / 0.47	34.35 / 0.47
25	34.21 / 0.47	34.24 / 0.47	34.07 / 0.47	34.26 / 0.47	34.23 / 0.47	34.20 / 0.47
P-ResNet with WP (estimated SNR / LLR)						
0	15.22 / 0.86	32.58 / 0.79	14.14 / 1.02	29.04 / 0.75	39.48 / 0.79	26.09 / 0.84
5	38.97 / 0.64	57.25 / 0.60	36.74 / 0.70	51.67 / 0.59	62.66 / 0.61	49.46 / 0.63
10	53.23 / 0.52	57.93 / 0.51	49.75 / 0.53	58.03 / 0.50	59.30 / 0.51	55.65 / 0.51
15	54.03 / 0.49	55.02 / 0.48	51.38 / 0.49	56.45 / 0.48	55.36 / 0.48	54.45 / 0.48
20	53.71 / 0.48	53.73 / 0.48	52.31 / 0.48	54.07 / 0.48	53.80 / 0.48	53.52 / 0.48
25	53.60 / 0.48	53.58 / 0.48	53.28 / 0.48	53.68 / 0.48	53.56 / 0.48	53.54 / 0.48
P-ResNet with UP (estimated SNR / LLR)						
0	10.65 / 0.86	20.78 / 0.82	11.41 / 0.99	20.74 / 0.77	22.91 / 0.83	17.30 / 0.85
5	24.49 / 0.65	34.68 / 0.62	23.54 / 0.69	31.38 / 0.59	38.09 / 0.62	30.44 / 0.63
10	38.09 / 0.54	40.87 / 0.53	34.71 / 0.54	40.11 / 0.51	42.19 / 0.52	39.19 / 0.53
15	41.02 / 0.50	41.27 / 0.50	38.70 / 0.50	42.06 / 0.49	41.65 / 0.50	40.94 / 0.50
20	40.53 / 0.49	40.41 / 0.49	39.40 / 0.49	40.56 / 0.49	40.50 / 0.49	40.28 / 0.49
25	40.26 / 0.49	40.25 / 0.49	40.02 / 0.49	40.31 / 0.49	40.28 / 0.49	40.22 / 0.49

without a significant increase of distortion. In conclusion, we can state that the more consistent architecture along this study is the P-ResNet with weighted progressive criterion. This system achieved a positive trade-off throughout the evaluated conditions while staying competitive along all the experiments performed. These architectures obtained good results in dereverberation and also in denoising, so these architectures are advisable in speech enhancement tasks.

Future work will further study the progressive strategy on additional DNN architectures such as U-Net and GAN. We will also assess the performance of 2D-convolutions, as the core of convolutional blocks, and compare them with 1D-convolutions.

Abbreviations

Adam: Adaptive moment estimator; AVG: Average; CNN: Convolutional neural network; DCT: Discrete cosine transform; DNN: Deep neural network; DNN-SE: Deep neural network speech enhancement; FB: Mel-scaled filter bank; LLR: Log-likelihood ratio; LSA: Log-spectral amplitude; LSTM: Long short-term memory; MFCC: Mel-frequency cepstral coefficients; MSE: Mean square error; P-CNN: Progressive convolutional neural network; PReLU: Parametric rectified linear unit; P-ResNet: Progressive residual neural network; PSE: Progressive speech enhancement; ResNet: Residual neural network; RIR: Room impulse responses; SE: Speech enhancement; SNR: Signal-to-noise ratio; SRMR: Speech-to-reverberation modulation energy ratio; STD: Standard deviation; STFT: Short-term fourier transform; UP: Uniform progressive; WP: Weighted progressive; WPE: Weighted prediction error

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.
This material is based upon work supported by Google Cloud.

Authors' contributions

JLL designed the PSE systems and performed the set of experiments, took an important role in the analysis of the results, and he was the main contributor in writing the manuscript. DR proposed the metrics, part of the experimental methodology, and took important part in manuscript writing. AM proposed the preliminary gradient analysis and guided the analysis of the results provided in this manuscript. LV helped in results analysis and results representation. AO and ELL helped to revise the manuscript and approved it for publication. The final manuscript was read and approved by all the authors.

Funding

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by the Government of Aragon (Reference Group T3617R) and co-financed with Feder 2014-2020 "Building Europe from Aragon".

Availability of data and materials

The Librispeech, Tedlium v2, and Musan datasets supporting the conclusions of this article are available in the openslr repository, http://openslr.org. The Timit, WSJCAM0, and MC-WSJ-AV datasets supporting the conclusions of this article are available in LDC repository, https://catalog.ldc.upenn.edu/LDC93S1W, https://catalog.ldc.upenn.edu/LDC95S24, and https://catalog.ldc.upenn.edu/LDC2014S03 respectively. WSJCAM0 and MC-WSJ-AV are used in REVERB Challenge for training, development and evaluation purposes. For more resources of REVERB challenge, visit download area in REVERB challenge web site, https://reverb2014.dereverberation.com . The VoiceHome datasets supporting the conclusions of this article are available in the Voice-Home repository, http://voice-home.gforge.inria.fr/voiceHome_corpus.html and http://voice-home.gforge.inria.fr/voiceHome-2_corpus.html. RIRs used in this work are generated with rir-generator, a python RIR generator library under GNU v3 License. This code is available in https://github.com/Marvin182/rir-generator.

Competing interests

The authors declare that they have no competing interests.

Received: 8 May 2020 Accepted: 2 December 2020
Published online: 07 January 2021

References

1. J. Llombart, D. Ribas, A. Miguel, L. Vicente, A. Ortega, E. Lleida, in *Proc. Interspeech 2019*. Progressive speech enhancement with residual connections, (Graz, 2019), pp. 3193–3197. https://doi.org/10.21437/Interspeech.2019-1748
2. T. Gao, J. Du, L. R. Dai, C. H. Lee, in *Interspeech 2016*. SNR-based progressive learning of deep neural network for speech enhancement, (San Francisco, 2016), pp. 3713–3717
3. T. Gao, J. Du, L. R. Dai, C. H. Lee, in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. Densely connected progressive learning for LSTM-based speech enhancement (IEEE, Alberta, 2018), pp. 5054–5058
4. M. H. Soni, N. Shah, H. A. Patil, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Time-frequency masking-based speech enhancement using generative adversarial network (IEEE, Alberta, 2018), pp. 5039–5043
5. H. S. Choi, J. H. Kim, J. Huh, A. Kim, J. W. Ha, K. Lee, in *International Conference on Learning Representations (ICLR 2018)*. Phase-aware speech enhancement with deep complex u-net, (Vancouver, 2018)
6. J. Abdulbaqi, Y. Gu, I. Marsic, RHR-Net: a residual hourglass recurrent neural network for speech enhancement. arXiv preprint arXiv:1904.07294 (2019). Accessed 06 July 2020
7. S. W. Fu, Y. Tsao, X. Lu, in *Interspeech 2016*. SNR-aware convolutional neural network modeling for speech enhancement, (San Francisco, 2016), pp. 3768–3772
8. S. R. Park, J. Lee, in *nterspeech 2017*. A fully convolutional neural network for speech enhancement, (Stockholm, 2017), pp. 1993–1997
9. H. Zhao, S. Zarar, I. Tashev, C. H. Lee, in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. Convolutional-recurrent neural networks for speech enhancement (IEEE, Alberta, 2018), pp. 2401–2405
10. Z. Chen, Y. Huang, J. Li, Y. Gong, in *Interspeech 2017*. Improving mask learning based speech enhancement system with restoration layers and residual connection, (Stockholm, 2017), pp. 3632–3637
11. J. Llombart, A. Miguel, A. Ortega, E. Lleida, in *IberSPEECH 2018*. Wide residual networks 1D for automatic text punctuation, (Barcelona, 2018), pp. 296–300
12. J. Llombart, D. Ribas, A. Miguel, L. Vicente, A. Ortega, E. Lleida, in *Proc. Interspeech 2019*. Speech enhancement with wide residual networks in reverberant environments, (Graz, 2019), pp. 1811–1815. https://doi.org/10.21437/Interspeech.2019-1745
13. L. Wyse, Audio spectrogram representations for processing with convolutional neural networks. arXiv preprint arXiv:1706.09559 (2017). Accessed 06 July 2020
14. S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, M. Gabbouj, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1-D convolutional neural networks for signal processing applications (IEEE, Brighton, 2019), pp. 8360–8364
15. B. Wu, K. Li, M. Yang, C. H. Lee, A reverberation-time-aware approach to speech dereverberation based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(1), 102–111 (2016)
16. A. Rousseau, P. Deléglise, Y. Esteve, in *LREC 2014*. Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks, (Reykjavik, 2014), pp. 3935–3939
17. V. Panayotov, G. Chen, D. Povey, S. Khudanpur, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Librispeech: an ASR corpus based on public domain audio books (IEEE, South Brisbane, 2015), pp. 5206–5210
18. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon Tech. Rep. n. **93**, 27403 (1993)
19. J. B. Allen, D. A. Berkley, Image method for efficiently simulating small-room acoustics. J. Acoust. Soc. Am. **65**(4), 943–950 (1979)
20. D. Snyder, G. Chen, D. Povey, MUSAN: a music, speech, and noise corpus. arXiv:1510.08484v1 (2015). http://arxiv.org/abs/1510.08484. Accessed 08 July 2020

21. K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, B. Raj, in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-13)*. The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech (IEEE, New Paltz, 2013), pp. 1–4
22. N. Bertin, E. Camberlein, E. Vincent, R. Lebarbenchon, S. Peillon, É. Lamandé, S. Sivasankaran, F. Bimbot, I. Illina, A. Tom, *et al*, in *Interspeech 2016*. A French corpus for distant-microphone speech processing in real homes, (San Francisco, 2016), pp. 2781–2785
23. N. Bertin, E. Camberlein, R. Lebarbenchon, E. Vincent, S. Sivasankaran, I. Illina, F. Bimbot. VoiceHome-2, an extended corpus for multichannel speech processing in real homes, vol. 106, (2019), pp. 68–78
24. T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition (IEEE, Detroit, 1995), pp. 81–84
25. M. Lincoln, I. McCowan, J. Vepa, H. K. Maganti, in *Proceedings of the 2005 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU-05)*. The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): specification and initial experiments (IEEE, Philadelphia, 2005), pp. 357–362
26. C. Kim, R. M. Stern, in *Ninth Annual Conference of the International Speech Communication Association (Interspeech 2008)*. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis, (Brisbane, 2008)
27. T. H. Falk, C. Zheng, W. Y. Chan, A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech. IEEE Trans. Audio Speech Lang. Process. **18**(7), 1766–1774 (2010)
28. J. F. Santos, M. Senoussaoui, T. H. Falk, in *Proc. Int. Workshop Acoust. Signal Enhancement (IWAENC 2014)*. An updated objective intelligibility estimation metric for normal hearing listeners under noise and reverberation, (Antibes - Jaun les Pins, 2014), pp. 55–59
29. P. C. Loizou, *Speech quality asssessment. in: multimedia analysis, processing and communications*. (Springer, Berlin, 2011), pp. 623–654
30. A. W. Rix, J. G. Beerends, M. P. Hollier, A. P. Hekstra, in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221), vol. 2*. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs (IEEE, Salt Lake City, 2001), pp. 749–752
31. L. Drude, J. Heymann, C. Boeddeker, R. Haeb-Umbach, *NARA-WPE: a Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing*, (Stuttgart, 2018), pp. 1–5
32. T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, B.-H. Juang. Speech dereverberation based on variance-normalized delayed linear prediction, vol. 18, (2010), pp. 1717–1731

Publisher's Note